dr inż. Adam Słota
mgr inż. Mateusz Domka
Instytut Technologii Maszyn i Automatyzacji Produkcji
Politechnika Krakowska

PROGRAMOWANIE SKOORDYNOWANYCH RUCHÓW ROBOTÓW NA PRZYKŁADZIE ROBOTÓW ABB I FANUC

W pracy przedstawiono charakterystykę rozwiązań: MultiMove firmy ABB oraz DUALARM, Multi arm control i Robot Link firmy Fanuc Robotics umożliwiających koordynację ruchów robotów. Przedstawiono przykłady konfiguracji stanowisk, wymagane opcje oprogramowania oraz kalibrację układów współrzędnych robotów. Dla rozwiązania MultiMove przedstawiono przykład stanowiska wykorzystującego koordynację ruchów do przenoszenia przedmiotu przez dwa roboty oraz zamieszczono fragmenty programów.

PROGRAMMING OF COORDINATED MOTION FOR ABB AND FANUC ROBOTS

In the paper description of functionalities of robots' motion coordination provided by ABB and Fanuc Robotics is presented - solutions MultiMove, DUALARM, Multi arm control and Robot Link are described. Examples of configuration, required software options and TCP calibration are presented. For MultiMove solution an example of work cell which uses coordinated motion of two robots for workpiece transport is presented. Sample robot programs are included.

1. WSTĘP

Udział operacji zautomatyzowanych realizowanych przez roboty przemysłowe w procesach produkcyjnych jest coraz większy. Zwiększa się również poziom złożoności takich operacji. Związane to jest z jednej strony ze wzrostem złożoności wytwarzanych produktów oraz wzrostem wymagań co do ich jakości i trwałości, a z drugiej strony z postępami w dziedzinie robotyki. Te dwa trendy występują równocześnie i wzajemnie silnie na siebie oddziałują.

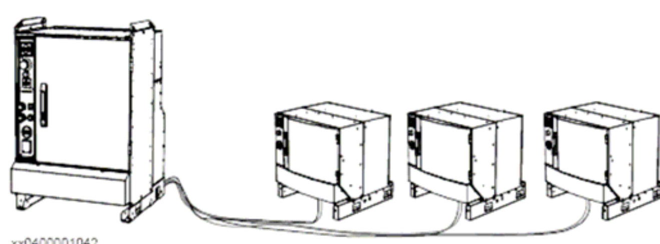
Roboty przemysłowe wykorzystywane są do realizacji paletyzacji, montażu, malowania, spawania czy obróbki ubytkowej. Ze względu na fakt, że wykonują one wybrane fragmenty procesu produkcyjnego wymagana jest koordynacja ich działania z innymi urządzeniami. Koordynacja taka może dotyczyć synchronizacji rozpoczynania/kończenia wybranych fragmentów programów, aby zapewnić bezkolizyjną realizację zadań w wymaganej kolejności. Istnieją również zadania, których realizacja wymaga ciągłej koordynacji ruchów realizujących ją robotów. Taka sytuacja występuje przy realizacji spawania, gdy ze względu na kształt spawanych elementów są one w trakcie realizacji spawania przemieszczane przez robota lub pozycjoner. Inny przykład to zadanie przenoszenia dużego i ciężkiego przedmiotu przez dwa roboty.

Postęp w technice napędowej i sterowania sprawił, że producenci robotów przemysłowych dostarczają rozwiązań umożliwiających ciągłą koordynację ruchów robotów przemysłowych i realizację wymienionych powyżej zadań. W pracy przestawiono charakterystykę rozwiązań w zakresie ciągłej koordynacji ruchów robotów dostarczanych przez firmy ABB oraz Fanuc Robotics.

2. KONFIGURACJA STANOWISK WYKORZYSTUJĄCYCH KOORDYNACJĘ RUCHÓW ROBOTÓW

2.1 Konfiguracja stanowisk wielorobrotowych z wykorzystaniem robotów firmy ABB

Programowanie skoordynowanych ruchów robotów firmy ABB jest możliwe dzięki funkcjonalności *MultiMove* [1]. Cel funkcjonalności *MultiMove* to sterowanie ruchami kilku robotów przy użyciu jednego kontrolera (jednostki sterującej). Jeden kontroler typu IRC5 może sterować ruchami czterech robotów, przy czym każdy z nich musi posiadać odrębny moduł sterowania silnikami.



Rys. 1. Kontroler w konfiguracji "jednokabinowej" z modułami sterowania silnikami [1]

Moduły silnikowe są połączone z kontrolerem przewodami sieci Ethernet oraz przewodami bezpieczeństwa (po jednym przewodzie Ethernet i bezpieczeństwa na każdy moduł sterowania silnikami). Jeden kontroler może obsłużyć do 6 zadań ruchu równocześnie: 4 dla robotów oraz 2 dla pomocniczych urządzeń pozycjonujących (przy czym jedno zadanie ruchu może obsłużyć tylko jednego robota lub do sześciu osi urządzenia pozycjonującego).

Opcja *MultiMove* pozwala między innymi na realizację stanowisk, w których:

- kilka robotów realizuje skoordynowane ruchy w jednym poruszającym się układzie współrzędnych
- jeden z robotów porusza obiektem, na którym pracują inne roboty
- kilka robotów wykonuje skoordynowane ruchy np. podnosząc ciężkie przedmioty.

Korzystanie z opcji *MultiMove* wymaga przygotowania sprzętu (kontrolera, okablowania oraz modułów sterowania napędami). Należy również skonfigurować system kontrolera IRC5 za pomocą programu *RobotStudio*. System umożliwiający obsługę więcej niż jednego robota tworzy się za pomocą 'kreatora' dobierając odpowiednie opcje. Najważniejsze z nich, ze względu na możliwość programowania stanowisk z wykorzystaniem kilku manipulatorów i koordynacji ich ruchów to:

604-1 MultiMove Coordinated - daje możliwość zaprogramowania ruchów skoordynowanych między robotami lub między robotami i pozycjonerami.

623-1 Multitasking - umożliwia uruchamianie kilku zadań jednocześnie; po włączeniu tej opcji kontroler może obsługiwać do 20 zadań równocześnie, przy czym maksymalnie sześć z nich może być zadaniami ruchu.

Włączenie powyższych opcji jest niezbędne dla programowania skoordynowanych ruchów robotów sterowanych kontrolerem IRC5. Dodatkowo, w zależności od typu stacji, wymagane są też inne opcje, pozwalające na korzystanie z cyfrowych wejść oraz wyjść, dające możliwość sterowania zewnętrznym pozycjonerem, czy pozwalające na wykorzystanie sprzętu spawalniczego. Poniżej przedstawiono niektóre z nich:

709-x DeviceNet **-** pozwala na wykorzystanie protokołu DeviceNet (opartego na standardzie sieci CAN) do komunikacji między kontrolerem a jednostkami modułów wejść/wyjść.

Logical Cross Connection **-** pozwala na szybkie operowanie kombinacjami wyjść oraz wejść cyfrowych; dzięki niej dana kombinacja stanów wejść lub wyjść automatycznie wywołuje

akcję (np. w postaci ustawienia lub zerowania danego wyjścia) bez konieczności jej programowania w języku RAPID.

Multiple Axis Positioner – pozwala na wykorzystanie w stacji oraz programowanie ruchu pozycjonera wieloosiowego.

Advanced RAPID - jest opcją pozwalającą poszerzyć aplikacje języka RAPID o dodatkowe zaawansowane funkcjonalności takie jak nadawanie aliasów wejściom/wyjściom, wyszukiwanie danych, zarządzanie parametrami systemu oraz inne.

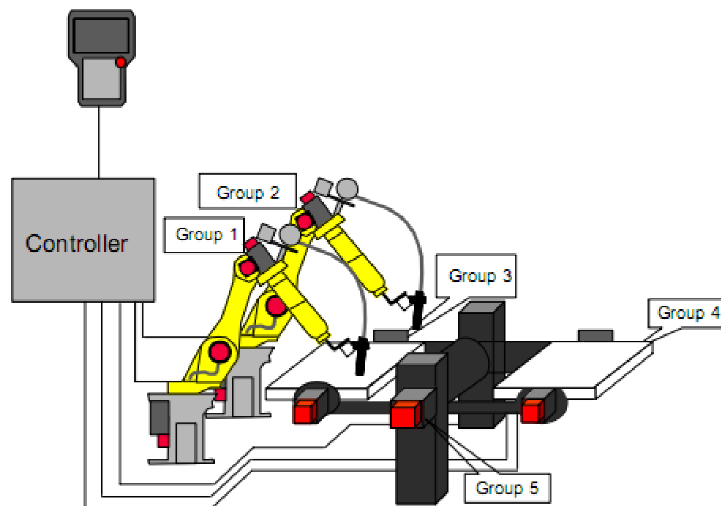
633-1 Arc – pozwala na wykorzystanie rozwiązań dedykowanych do procesów spawania.

2.2 Konfiguracja stanowisk wielorobrotowych z wykorzystaniem robotów firmy Fanuc Robotics

Rozwiązania firmy Fanuc Robotics umożliwiające realizację skoordynowanych ruchów robotów to:

- *DUALARM* [5]
- *Multi arm control* [3]
- *Robot Link* [4].

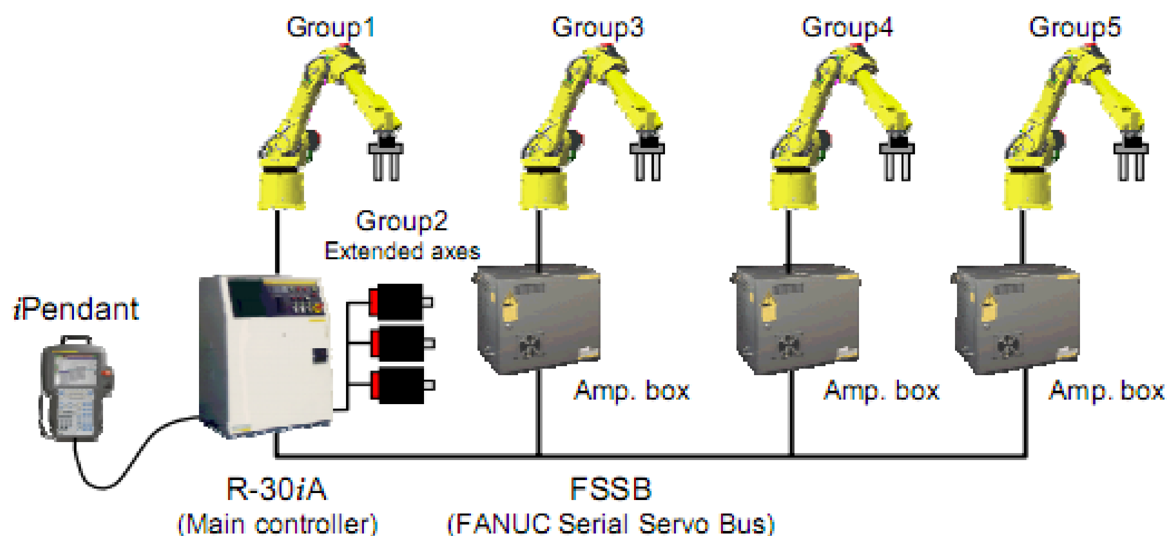
Funkcjonalności *DUALARM* oraz *Multi arm control* są rozwiązaniami podobnymi do tych proponowanych przez firmę ABB. Przy ich użyciu do jednego kontrolera można podłączyć maksymalnie dwa roboty (rozwiązanie *DUALARM*) lub cztery roboty (rowiązanie *Multi arm control*) i sterować nimi wykorzystując koordynację ruchów. Na rys. 2 przedstawiono schemat połączeń dwóch robotów i pozycjonera trzyosiowego dla opcji *DUALARM*.



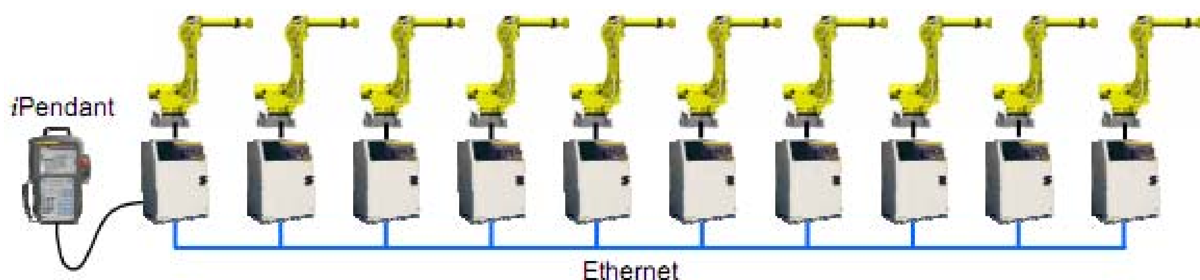
Rys. 2. Przykład połączenia robotów Fanuc przy wykorzystaniu opcji *DUALARM* [3]

Rozwiązanie *Multi arm control* pozwala na podłączenie do jednego kontrolera R-30iA do 40 sterowanych osi w 6-ciu grupach (przy czym maksymalna liczba robotów to 4). Wszystkie urządzenia mogą być sterowane przy pomocy jednego iPendant'a (rys. 3).

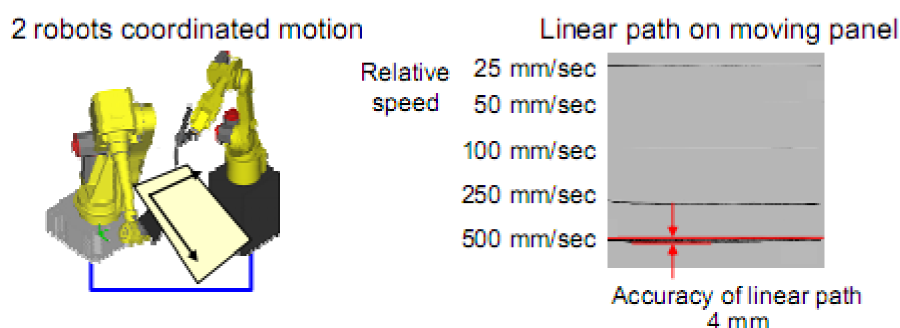
Rozwiązanie *Robot link* różni się od rozwiązań *DUALARM*, *Multi arm control* oraz rozwiązania *MultiMove* dostarczanego przez ABB tym, że do sterowania ruchami połączonych robotów potrzeba tyle kontrolerów ile jest robotów. Układy sterowania robotów połączone są siecią Ethernet. Funkcja *Robot link* pozwala na koordynację ruchów dla maksymalnie dziesięciu robotów i sterowanie oraz programowanie ruchów przy pomocy jednego iPendant'a.



Rys. 3. Schemat połączenia robotów Fanuc w rozwiązaniu *Multi arm control* [2]



Rys. 4. Schemat połączenia dla rozwiązania *Robot link* [2]



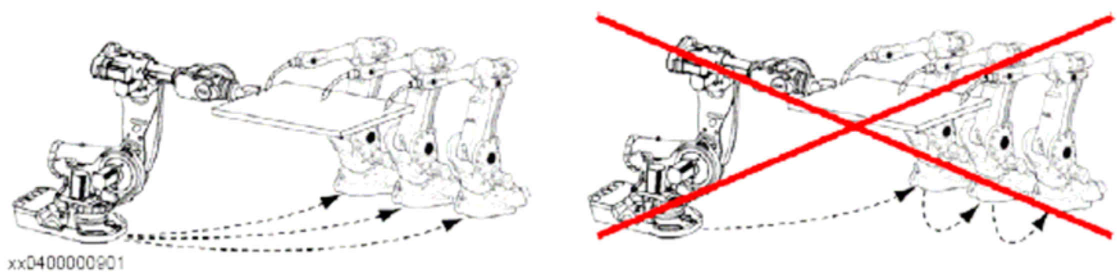
Rys. 5. Wpływ szybkości ruchów na dokładność dla rozwiązania *Robot link* [2]

Ze względu na sterowanie poszczególnych robotów przez własne kontrolery dokładność odwzorowania ścieżki ruchu robotów dla rozwiązania *Robot link* zależna jest od prędkości realizowanych ruchów. Na rys. 5 przedstawiono wpływ prędkości ruchu na uzyskiwaną dokładność ruchu dla dwóch robotów połączonych za pomocą opcji *Robot link*. Jeden robot trzyma płaski panel, natomiast drugi wykonuje ruchy w układzie współrzędnych poruszającym się wraz z panelem. Dla większych prędkości panelu, ścieżki ruchu wykonywanego w jego układzie współrzędnych są mniej dokładne (dla prędkości 500 mm/s występująca niedokładność to 4 mm).

2.3 Kalibracja robotów i dokładność ruchów

W przypadku gdy w stanowisku zrobotyzowanym wykorzystuje się więcej niż jednego robota, należy odpowiednio skalibrować układy współrzędnych robotów, aby możliwe było wzajemnie jednoznacznie określenie położenie punktów TCP w przestrzeni roboczej.

U obu producentów (ABB oraz Fanuc Robotics) sposób kalibracji jest podobny. Dokonuje się jej najlepiej względem jednego robota. Należy pamiętać, aby unikać tzw. ”łańcucha kalibracji” – sytuację taką przedstawia rys. 6.



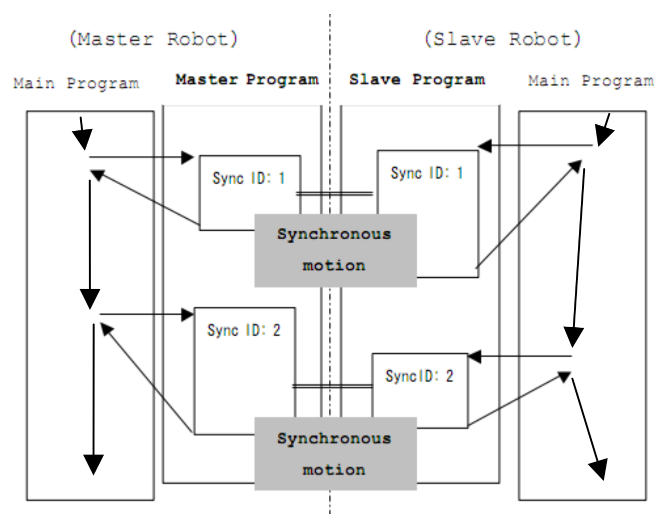
Rys. 6. Przykłady poprawnej i niepoprawnej kalibracji [1]

Łańcuch kalibracji jest niekorzystny ze względu na sumowanie się błędu kalibracji w miarę jak dokonuje się jej kolejno w odniesieniu do poprzedniego robota. W przykładzie kalibracji niepoprawnej przedstawionej na rys. 6 robot pierwszy z lewej (z robotów spawalniczych) będzie miał najdokładniej skalibrowany układ współrzędnych, najmniej dokładnie będzie skalibrowany układ ostatniego robota na prawo, gdyż będzie on uwzględniał błędy poprzednich dwóch robotów z łańcucha oraz własny błąd kalibracji.

3. PROGRAMOWANIE DZIAŁAŃ SKOORDYNOWANYCH NA PRZYKŁADZIE ROBOTÓW FANUC ROBOTICS ORAZ ABB

3.1 Idea programowania ruchów skoordynowanych – Fanuc Robotics

Idea synchronizacji programów oraz programowania ruchów skoordynowanych u obu producentów jest podobna. W przypadku urządzeń firmy Fanuc mamy do czynienia z podziałem na robota MASTER oraz roboty SLAVE. Koordynacja działań jest osiągana przez śledzenie przez roboty SLAVE ruchów robota MASTER.



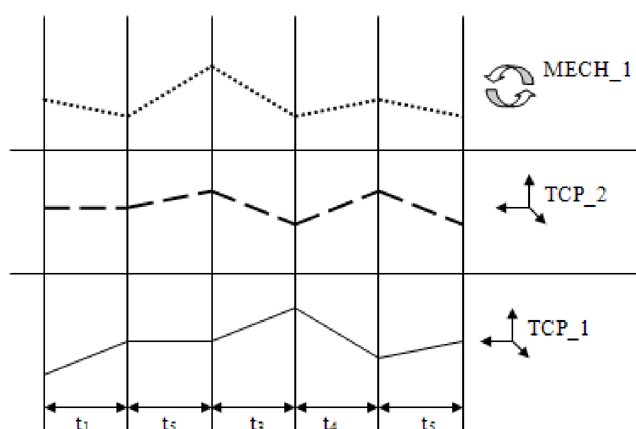
Rys. 7. Idea działania programów robotów o skoordynowanych ruchach (strzałki obrazują sposób wykonywania programu przez kontroler robota) [4]

Dla układu dwóch robotów kontroler wykonuje dwa programy równocześnie. W momencie gdy programy dojdą do miejsca, w którym następuje synchronizacja (oznaczonego specjalnym identyfikatorem synchronizacji) robot SLAVE rozpoczyna śledzenie ruchów robota MASTER.

3.2 Idea programowania ruchów skoordynowanych – ABB

Programowanie robotów z wykorzystaniem opcji *MultiMove* firmy ABB odbywa się przy użyciu odpowiednich komend rozpoczynających i kończących synchronizację ruchów. Ruch zsynchronizowany zaczyna się w momencie, gdy pointer wykonywania programu napotka odpowiednią komendę: *SyncMoveOn* – rozpoczęcie synchronizacji ruchów. Synchronizację ruchów kończy znacznik: *SyncMoveOff*. Dodatkowo wykorzystuje się komendę *WaitSyncTask* odpowiadającą za synchronizację programów w miejscu jej użycia. Wszystkie komendy znajdujące się poza znacznikami synchronizacji wykonywane są niezależnie i bez synchronizacji. Podczas programowania należy pamiętać o tym, że pomiędzy znacznikiem początku i końca synchronizacji ruchu musi się znajdować dokładnie tyle samo komend ruchu oraz ich numery ID muszą się zgadzać ze sobą.

Ważną cechą systemów wykorzystujących opcję *MultiMove* jest czas wykonywania poszczególnych zsynchronizowanych ruchów. Po rozpoczęciu synchronizacji ruchów każda z instrukcji ruchu czy to robota czy urządzenia pomocniczego (np. pozycjonera) jest wykonywana w tym samym czasie.



Rys. 8. Ilustracja wymagań dotyczących czasów ruchów punktów TCP przy realizacji ruchów zsynchronizowanych ($t_1 = t_2 = t_3 = t_4 = t_5$)

Prędkości ruchu robotów w zsynchronizowanych zadaniach mogą się różnić od wartości zaprogramowanych w komendach ruchu. Jeżeli roboty, których ruchy zostały zsynchronizowane mają ustawione w komendach ruchu te same prędkości a różne dystanse do pokonania dla ich punktów TCP to prędkość robota mającego do wykonania krótsze przemieszczenie zostanie zmniejszona. Wynika to z konieczności zachowania równych czasów ruchu punktów TCP dla wszystkich komend ruchów znajdujących się pomiędzy komendami synchronizacji.

Jak wspomniano wcześniej liczba komend ruchu musi być taka sama między komendami synchronizacji (*SyncMoveOn, SyncMoveOff*). Może jednak wystąpić sytuacja gdy liczba komend ruchu ramienia manipulatora zależy od czynników zewnętrznych (najczęściej sygnałów z cyfrowych wejść lub na przykład z systemu wizyjnego), wtedy stosuje się tzw.

instrukcje ‘dummy’. Wstawienie takiej instrukcji polega na skopiowaniu wcześniej występującej instrukcji ruchu i nadaniu jej następnego w kolejności identyfikatora ID.

Przykład:

```
PERS tasks sync_tasks{2} := [["T_PROCROB"],["T_HANDLEROB"]];
VAR syncident sync1;
VAR syncident sync2;
SyncMoveOn sync1, sync_tasks;
MoveL p1\ID:=10, v100, z20, tool0 \WObj:=wobj0;
IF ISignalDI di1=1 THEN MoveL p2\ID:=20, v150, z20, tool1 \WObj:= wobj0;
    MoveL p1\ID:=30, v150, z20, tool1 \WObj:= wobj0;
ELSE  MoveL p1\ID:=20, v150, z20, tool1 \WObj:= wobj0;
MoveL p1\ID:=30, v150, z20, tool1 \WObj:= wobj0;
ENDIF
SyncMoveOff sync2;
```

Pogrubionym drukiem została zaznaczona instrukcja dodana w celu zachowania tej samej liczby instrukcji poddanych synchronizacji. Jeżeli wejściowy sygnał *di1* jest w stanie wysokim, robot wykona ruch do punktu *p2* a następnie powróci do punktu *p1*, jeżeli wejście jest w stanie niskim – robot pozostanie w bezruchu w punkcie *p1*. W ten sposób, jeżeli w innych zsynchronizowanych zadaniach występują 2 różne komendy ruchu nie wystąpi żaden błąd gdyż zachowana zostanie liczba instrukcji między komendami początku i końca synchronizacji.

Podczas programowania stacji zrobotyzowanej z wykorzystaniem koordynacji działań manipulatorów i urządzeń zewnętrznych istotny jest również sposób deklarowania układów współrzędnych urządzeń. W przypadku gdy wykorzystywany jest pozycjoner (może nim być również jeden z robotów) należy z nim związać układ współrzędnych, w którym definiowane będą ruchy innych urządzeń, np. robotów spawalniczych.

PERS wobjdata wobj_mech_1 :=[***FALSE***, ***FALSE***, ***"MECH_1"***,*[[0, 0, 0], [0 0, 0 ,1]],[[0, 0,880 [1, 0, 0, 0]]];*

Opcja ‘robhold’ definiuje czy robot trzyma obiekt manipulacji w danym zadaniu.

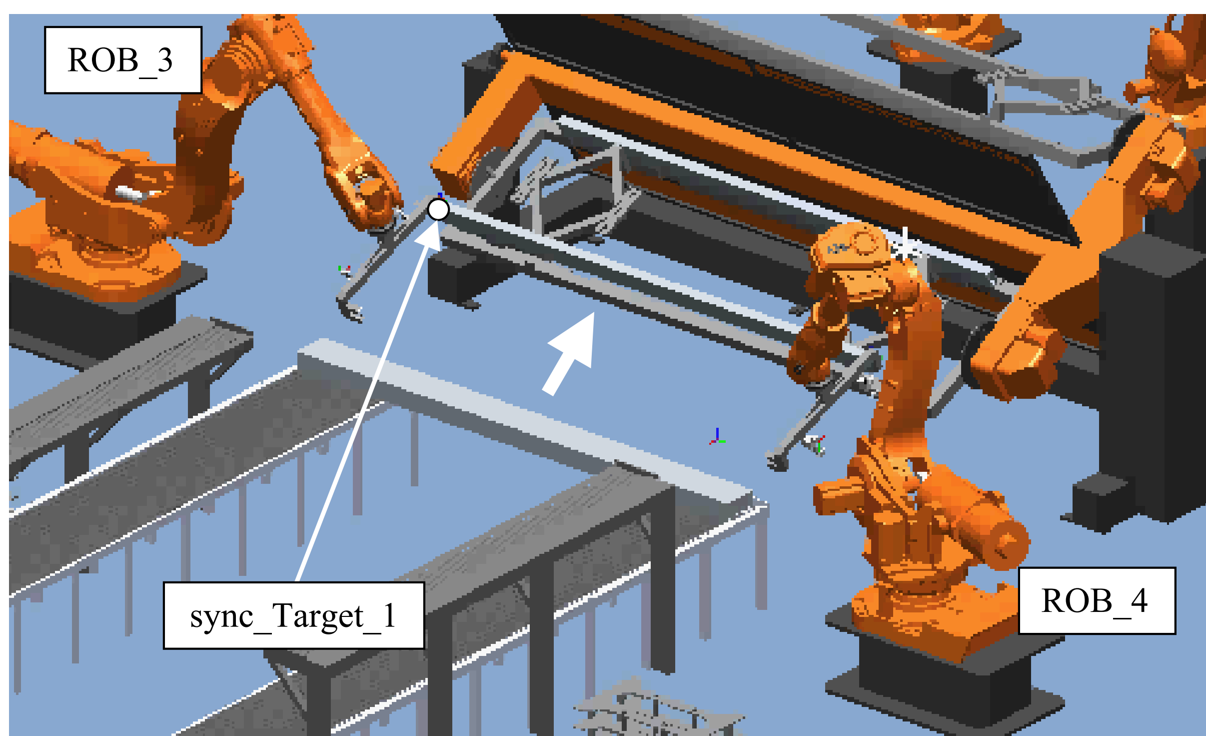
Opcja ‘ufprog’ definiuje czy obiekt manipulacji jest stacjonarny (‘true’) jeśli jest on ruchomy opcja jest ustawiona na ‘false’

Opcja ‘ufmech’ to nazwa zewnętrznej osi pozycjonera mechanicznego, z którym związany jest układ współrzędnych

Gdy mamy do czynienia z koordynacją ruchów dwóch robotów w celu przeniesienia jednego obiektu sytuacja wygląda podobnie. Trzeba związać układ współrzędnych z jednym z robotów, a następnie po rozpoczęciu synchronizacji ruchów (gdy przedmiot manipulacji jest uchwycony) zaprogramować ruch drugiego robota do zadanego punktu w ruchomym układzie współrzędnych.

3.3 Przykład programowania ruchów skoordynowanych na podstawie robotów firmy ABB

Przedstawiony poniżej przykład został zrealizowany w oprogramowaniu *RobotStudio*. W przykładowej stacji wykorzystano dwa roboty IRB6650S, których zadaniem jest załadunek i rozładunek pozycjonera spawalniczego. Roboty wspólnie przenoszą stalowe podłużnice ramy na pozycjoner spawalniczy trójosiowy oraz później całą zespawaną ramę (przy rozładunku) na podajnik rolkowy. W przykładzie zostały wykorzystane możliwości jakie daje opcja *MultiMove* firmy ABB.



Rys. 9. Roboty ABB podczas załadunku stalowej podłużnicy na pozycjoner spawalniczy

Gdy roboty nie przenoszą elementów, ich ruchy zaprogramowane są w nieruchomych układach współrzędnych. W momencie gdy następuje uchwycenie podłużnicy albo całej ramy, włączana jest synchronizacja ruchów i robot ROB_3 zaczyna śledzić punkt ('sync_Target_1' w listingu przedstawionym poniżej) znajdujący się w układzie współrzędnych ruchomym – związanym z robotem ROB_4.

Poniżej przedstawiono przykładowe procedury programów robotów przenoszących ramę, w których zachodzi pełna koordynacja ruchów oraz synchronizacja między nimi. Zaobserwować można, że robot ROB_3 działa podczas synchronizacji w układzie 'mov_wobj' (jest to układ poruszany przez robota ROB_4). Ruchy robota ROB_4 realizowane są w układzie nieruchomym – związanym na przykład z podajnikiem.

Procedura robota ROB_3:

PROC mload_stringers()

MoveJ m_home_2,v1000,z100,irb_2_gripper_stringer\WObj:=conveyor;

WaitDO POS_IN_START,1;

…

Set ROB3_IGRIP;

WaitTime 0.5;

SyncMoveOn sync1, handling_tasks; *←początek ruchów skoordynowanych A*

MoveL sync_Target_1\ID:=1,v1000,fine,irb_2_gripper_stringer\WObj:=mov_wobj;

MoveL sync_Target_1\ID:=2,v1000,fine,irb_2_gripper_stringer\WObj:=mov_wobj;

MoveL sync_Target_1\ID:=3,v1000,fine,irb_2_gripper_stringer\WObj:=mov_wobj;

Reset ROB3_IGRIP;

WaitTime 0.5;

SyncMoveOff sync2*; ← zakończenie koordynacji A*

MoveL m_Target_70,v500,fine,irb_2_gripper_stringer\WObj:=pos_holder;

…

Set ROB3_IGRIP;

WaitTime 0.5;

SyncMoveOn sync3, handling_tasks; *←początek ruchów skoordynowanych B*

MoveL sync_Target_1\ID:=4,v1000,fine,irb_2_gripper_stringer\WObj:=mov_wobj;

MoveL sync_Target_1\ID:=6,v1000,fine,irb_2_gripper_stringer\WObj:=mov_wobj;

MoveL sync_Target_1\ID:=7,v1000,fine,irb_2_gripper_stringer\WObj:=mov_wobj;

Reset ROB3_IGRIP;

WaitTime 0.5;

SyncMoveOff sync4; *← zakończenie koordynacji B*

ENDPROC

Procedura robota ROB_4:

PROC load_stringers()

Set BLOCKADE;

MoveJ home,v1000,z100,irb_3_gripper_stringer\WObj:=conveyor;

…

Set ROB4_IGRIP;

WaitTime 0.5;

SyncMoveOn sync1, handling_tasks; *←początek ruchów skoordynowanych A*

MoveL Target_40\ID:=1,v1000,fine,irb_3_gripper_stringer\WObj:=pos_holder;

MoveL Target_50\ID:=2,v1000,fine,irb_3_gripper_stringer\WObj:=pos_holder;

MoveL Target_60\ID:=3,v500,fine,irb_3_gripper_stringer\WObj:=pos_holder;

Reset ROB4_IGRIP;

```
Set CONVEY;
WaitTime 0.5;
SyncMoveOff sync2; ← zakończenie koordynacji A
MoveL Target_70,v500,fine,irb_3_gripper_stringer\WObj:=pos_holder;
...
Set ROB4_IGRIP;
WaitTime 0.5;
SyncMoveOn sync3, handling_tasks; ←początek  ruchów skoordynowanych B
MoveL Target_40\ID:=4,v1000,fine,irb_3_gripper_stringer\WObj:=pos_holder;
MoveL Target_90\ID:=6,v500,fine,irb_3_gripper_stringer\WObj:=pos_holder;
MoveL Target_100\ID:=7,v500,fine,irb_3_gripper_stringer\WObj:=pos_holder;
Reset ROB4_IGRIP;
WaitTime 0.5;
SyncMoveOff sync4; ← zakończenie koordynacji B
ENDPROC
```

Koordynacja oznaczona w listingu jako *A* odpowiada za przeniesienie na pozycjoner spawalniczy pierwszej z dwóch podłużnic, natomiast synchronizacja oznaczona jako *B* – dotyczy przeniesienia podłużnicy drugiej. W powyższym listingu częściowo pominięto (wykropkowano) komendy w niezsynchronizowanych obszarach programu. Pozostawiona część programu obrazuje w pełni jego sens oraz sposób synchronizacji ruchów robotów.

4. PODSUMOWANIE

Dostarczane przez producentów robotów rozwiązania z zakresu koordynacji ruchów robotów w większości przypadków realizowane są w oparciu o sterowanie dwóch lub większej liczby robotów (i urządzeń pomocniczych) przez jeden kontroler. Rozwiązanie takie eliminuje konieczność korzystania z sygnałów wejść/wyjść w celu koordynacji realizacji programów. Skoordynowane ruchy robotów i urządzeń pomocniczych zarządzane są przez pojedynczy moduł planowania ruchu (*motion planner*). Ułatwia to integrację robotów i urządzeń pomocniczych stanowiska zrobotyzowanego W pracy przedstawiono charakterystykę rozwiązań: *MultiMove* firmy ABB oraz *DUALARM*, *Multi arm control* oraz *Robot Link* firmy Fanuc Robotics.

5. LITERATURA

[1] MultiMove 3HAC021272-001_RevG_en.pdf, ABB**,** *Application manual - MultiMove*

[2] http://www.innovationgrid.org/presentation/Auto_Robo_Expo_presentation.pdf

[3] R-J3iC_MultiArm_[R-J3iC_HandlingTool_Operator_ManualMAROCH70008051E REV.A].pdf, Fanuc Robotics

[4] R-30iA_RobotLink_manual_[B-82924EN_01].pdf, Fanuc Robotics

[5] http://www.dinsaonline.com/aplicaciones/pdf/dualram_system/dualarmSystem.pdf